What is claimed is:

1. A virtual machine that executes a virtual machine instruction sequence under control of a real machine, comprising:

stack means for temporarily storing data in a last-in first-out format;

instruction storing means for storing the virtual machine instruction sequence and a plurality of sets of succeeding instruction information, wherein each virtual machine instruction in the virtual machine instruction sequence is associated with a set of succeeding instruction information that indicates a change in a storage state of the data in the stack means due to execution of a virtual machine instruction executed after the associated virtual machine instruction;

read means for reading a virtual machine instruction and an associated set of succeeding instruction information from the instruction storing means; and

decoding-executing means for specifying and executing operations corresponding to a combination of the read virtual machine instruction and the read set of succeeding instruction information.

2. The virtual machine of Claim 1, wherein the decoding-executing means includes:

a real machine instruction sequence storing unit for

storing a plurality of real machine instruction sequences that correspond to all combinations of virtual machine instructions and sets of succeeding instruction information;

a specifying unit for specifying a real machine instruction sequence in the real machine instruction sequence storing unit, the real machine instruction sequence corresponding to a combination of the virtual machine instruction and the set of succeeding instruction information read by the read means; and

an executing unit for executing the specified real machine instruction sequence.

3. The virtual machine of Claim 2, wherein each set of succeeding instruction information indicates a change in a number of sets of data in the stack means due to execution of a virtual machine instruction executed after a virtual machine instruction associated with the set of succeeding instruction information and

wherein at least one real machine instruction sequence stored in the real machine instruction sequence storing unit contains real machine instructions that perform a stack handling in the stack means in advance for a virtual machine instruction that is to be executed based on a set of succeeding instruction information associated with a currently executed virtual machine instruction.

4. The virtual machine of Claim 3, wherein the real machine instruction sequences stored in the real machine instruction sequence storing unit are composed with a premise that regions of the stack means used to store two sets of data to be read first and second are mapped to two registers in the real machine.

5. The virtual machine of Claim 1, wherein the instruction storing means includes a first storage area for storing the virtual machine instruction sequence and a second storage area for storing the sets of succeeding instruction information, wherein each location that stores a virtual machine instruction in the first storage area is associated with a location that stores an associated set of succeeding instruction information in the second storage area and

wherein the read means reads the virtual machine instruction from a location in the first storage area and the associated set of succeeding instruction information from a location in the second storage area, the location in the first storage area being associated with the location in the second storage area.

6. The virtual machine of Claim 1, wherein the virtual machine instruction sequence stored in the instruction storing means is an extended virtual machine instruction sequence that includes extended virtual machine

instructions, the extended virtual machine instructions being combinations of virtual machine instructions and associated sets of succeeding instruction information,

wherein the read means reads an extended virtual machine instruction from the instruction storing means, and

wherein the decoding-executing means specifies and executes operations corresponding to the extended virtual machine instruction.

7. A compiler that generates programs for a virtual machine with a stack architecture that includes a stack, comprising:

instruction sequence converting means for converting a source program into a virtual machine instruction sequence executable by the virtual machine;

succeeding instruction information generating means for generating sets of succeeding instruction information corresponding to virtual machine instructions in the virtual machine instruction sequence, each set of succeeding instruction information indicating a change in a storage state of data in the stack due to execution of a virtual machine instruction executed after a virtual machine instruction corresponding to the set of succeeding instruction information; and

associating means for associating each set of generated succeeding instruction information with a

corresponding virtual machine instruction and outputting the set of succeeding instruction information and the virtual machine instruction.

8. A virtual machine that executes a virtual machine instruction sequence under control of a real machine, comprising:

instruction storing means for storing the virtual machine instruction sequence;

read means for reading a virtual machine instruction in the virtual machine instruction sequence from the instruction storing means; and

decoding-executing means for specifying and executing operations corresponding to the virtual machine instruction,

wherein the decoding-executing means includes

a branch instruction judging unit for judging if the virtual machine instruction is a branch instruction and

an interrupt handling unit for detecting, if the virtual machine instruction is judged to be a branch instruction, whether there is an interrupt request, and, if so, performing a corresponding interrupt handling in addition to executing the branch instruction.

9. The virtual machine of Claim 8, wherein the decoding-executing means further includes

a real machine instruction sequence storing unit for

storing real machine instruction sequences corresponding to every virtual machine instruction and real machine instruction sequences for having interrupt handling performed corresponding to each interrupt request and

an executing unit for executing a real machine instruction sequence corresponding to the virtual machine instruction read by the read means,

wherein if the virtual machine instruction is judged to be the branch instruction and an interrupt request is detected, the interrupt handling unit has the executing unit execute a real machine instruction sequence for having the corresponding interrupt handling performed and then the real machine instruction sequence corresponding to the branch instruction.

10. A virtual machine that executes a virtual machine instruction sequence under control of a real machine, comprising:

instruction storing means for storing the virtual machine instruction sequence;

read means for reading a virtual machine instruction in the virtual machine instruction sequence from the instruction storing means; and

decoding-executing means for specifying and executing operations corresponding to the read virtual machine instruction,

wherein the decoding-executing means includes

a block judging unit for judging if the read virtual machine instruction is a virtual machine instruction representative of a block, a block being a predetermined number of virtual machine instructions and

an interrupt handling unit for detecting, if the read virtual machine instruction is judged to be the representative virtual machine instruction, whether there is an interrupt request to the virtual machine, and if so, performing a corresponding interrupt handling in addition to executing the representative virtual machine instruction.

11. The virtual machine of Claim 10, wherein the decoding-executing means includes

a real machine instruction sequence storing unit for storing a plurality of real machine instruction sequences corresponding to every virtual machine instruction and at least one real machine instruction sequence for having interrupt handling performed in response to an interrupt request and

an executing unit for executing a real machine instruction sequence corresponding to the read virtual machine instruction,

wherein the block judging unit judges that the read virtual machine instruction is a virtual machine instruction representative of the block when a number of virtual machine

instructions that have been read is equal to a multiple of the predetermined number and

wherein if the read virtual machine instruction is judged to be a representative virtual machine instruction and an interrupt request has been detected, the interrupt handling unit has the executing unit execute a real machine instruction sequence for having the interrupt handling performed and then the real machine instruction sequence corresponding to the representative virtual machine instruction.

12. A virtual machine that executes a virtual machine instruction sequence under control of a real machine, comprising:

real machine program storing means for storing a plurality of subprograms composed of real machine instructions;

instruction storing means that includes a first area for storing the virtual machine instruction sequence and a second area for storing a plurality of pointers to the subprograms in the real machine program storing means;

read means for reading a virtual machine instruction in the virtual machine instruction sequence from the first area in the instruction storing means; and

decoding-executing means for specifying and executing operations corresponding to the read virtual machine

instruction,

wherein the decoding-executing means includes

an area judging unit for judging whether the virtual machine instruction is an instruction that transfers control flow to a location in the second area and

an address converting-executing unit for executing, if the virtual machine instruction is judged to be an instruction that transfers control flow to a location in the second area, a subprogram indicated by a pointer stored in the location.

13. The virtual machine of Claim 12, wherein the first area and the second area in the instruction storing means are two adjacent storage areas whose boundary is marked by an address and

wherein the area judging unit judges, when the read virtual machine instruction is a call instruction for a subprogram, whether the virtual machine instruction is an instruction that transfers control flow, by comparing a call address of the call instruction with the address.

14. A virtual machine that executes a virtual machine instruction sequence under control of a real machine, comprising:

instruction storing means for storing the virtual machine instruction sequence;

read means for reading a virtual machine instruction in the virtual machine instruction sequence from the instruction storing means; and

decoding-executing means for specifying and executing operations corresponding to the read virtual machine instruction,

wherein the instruction storing means is a plurality of instruction blocks that constitute the virtual machine instruction sequence, the instruction blocks corresponding to basic blocks,

wherein the instruction blocks each include: an identifier area for storing an identifier that specifies a start position of the instruction block in the instruction storing means; a non-branch instruction area for storing non-branch instructions belonging to a corresponding basic block; and a branch instruction area for storing at least one branch instruction belonging to the corresponding basic block,

wherein each branch instruction stored in the branch instruction area designates a branch destination using an identifier stored in one of the identifier areas, and

wherein if the read virtual machine instruction is a branch instruction, the decoding-executing means has control flow branch to a start position of a non-branch instruction area in an instruction block having an identifier designated by the branch instruction as a branch destination.

15. The virtual machine of Claim 14, wherein the decoding-executing means includes a program counter composed of (a) an identifier register for storing an identifier of an instruction block to which a virtual machine instruction to be read belongs and (b) an offset counter for storing an offset that indicates a relative storage position of the virtual machine instruction in the instruction block,

wherein the read means reads the virtual machine instruction based on the identifier and the offset in the program counter,

wherein the decoding-executing means updates, if the read virtual machine instruction is the branch instruction, the program counter by writing the identifier designated as the branch destination by the branch instruction into the identifier register and by setting an initial value in the offset counter, and if the read virtual machine instruction is a non-branch instruction, updates the program counter by incrementing the offset counter, and

wherein the read means reads a virtual machine instruction to be executed next based on the program counter updated by the decoding-executing means.

16. The virtual machine of Claim 15, wherein the decoding-executing means includes a real machine instruction sequence storing unit that stores a plurality of real machine

instruction sequences that each correspond to a different virtual machine instruction,

wherein the instruction blocks in the instruction storing means each include a decoded data sequence area for storing a decoded data sequence that specifies real machine instruction sequences in the real machine instruction sequence storing unit, the real machine instruction sequences corresponding to virtual machine instructions stored in the non-branch instruction area and the branch instruction area of the instruction block,

wherein if a decoded data sequence is stored in an instruction block where reading is to be performed, the read means reads a set of decoded data in the decoded data sequence instead of a virtual machine instruction, and if not, the read means reads the virtual machine instruction and then generates a set of decoded data to specify a real machine instruction sequence in the real machine instruction sequence storing unit that corresponds to the virtual machine instruction, and

wherein the decoding-executing means reads from the real machine instruction sequence storing unit the real machine instruction sequence specified by the set of decoded data that has been either read or generated by the read means, and executes the real machine instruction sequence.

17. The virtual machine of Claim 16, wherein the decoded

data sequence area in the instruction storing means includes a flag area for storing a flag that indicates whether the decoded data sequence is stored in the decoded data sequence area,

wherein the decoding-executing means includes a current flag storing unit for storing a flag that is read from a flag area in a branch destination instruction block by the decoding-executing means when executing a branch instruction, and

wherein the read means reads a set of decoded data or a virtual machine instruction depending on the flag in the current flag storing unit.

18. The virtual machine of Claim 16, wherein each instruction block in the instruction storing means further includes a flag area for storing a flag that indicates whether a decoded data sequence is stored in the decoded data sequence area of the instruction block and

wherein the decoding-executing means includes a decoded data sequence writing unit for judging, after a branch instruction has been executed, whether the instruction block designated as the branch destination by the branch instruction stores a decoded data sequence by referring to a flag stored in a flag area of the instruction block, and if no decoded data sequence is stored, having a virtual machine instruction sequence in the instruction

block read, decoding the read virtual machine instruction sequence to produce a decoded data sequence, and writing the decoded data sequence into a decoded data sequence area in the instruction block.

19. A virtual machine that executes a virtual machine instruction sequence under control of a real machine, comprising:

instruction storing means for storing a compressed virtual machine instruction sequence to be executed;

read means for reading a compressed virtual machine instruction in the compressed virtual machine instruction sequence from the instruction storing means and decompressing the compressed virtual machine instruction to generate a decompressed virtual machine instruction; and

decoding-executing means for specifying and executing operations corresponding to the decompressed virtual machine instruction,

wherein the instruction storing means is a plurality of instruction blocks containing compressed virtual machine instructions constituting the compressed virtual machine instruction sequence, the instruction blocks corresponding to basic blocks,

wherein the instruction blocks each include: an identifier area for storing an identifier that specifies a start position of the instruction block in the instruction

storing means; a non-branch instruction area for storing compressed non-branch instructions belonging to a corresponding basic block; and a branch instruction area for storing at least one compressed branch instruction belonging to the corresponding basic block,

wherein each compressed branch instruction stored in a branch instruction area designates a branch destination using an identifier stored in one of the identifier areas, and

wherein if the decompressed virtual machine instruction is a branch instruction, the decoding-executing means has control flow branch to a start position of a non-branch instruction area in an instruction block having an identifier designated by the branch instruction as a branch destination.

20. The virtual machine of Claim 19, wherein each instruction block includes a decompression table area for storing a decompression table for use during decompression of compressed virtual machine instructions in the instruction block, the decompression table containing at least one combination of a compressed virtual machine instruction stored in the instruction block and a corresponding decompressed virtual machine instruction and

wherein the read means reads the compressed virtual machine instruction from the instruction storing means and

decompresses the compressed virtual machine instruction by referring to a decompression table in an instruction block to which the compressed virtual machine instruction belongs to generate the decompressed virtual machine instruction.

21. The virtual machine of Claim 20, wherein the decoding-executing means includes a program counter composed of (a) an identifier register for storing an identifier of an instruction block to which a compressed virtual machine instruction to be read belongs and (b) an offset counter for storing an offset that indicates a relative storage position of the compressed virtual machine instruction in the instruction block,

wherein the read means reads the compressed virtual machine instruction based on the identifier and the offset in the program counter,

wherein if the decompressed virtual machine instruction is a branch instruction, the decoding-executing means updates the program counter by writing the identifier designated as the branch destination by the branch instruction into the identifier register and by setting an initial value in the offset counter, and if the decompressed virtual machine instruction is a non-branch instruction, updates the program counter by incrementing the offset counter, and

wherein the read means reads a compressed virtual

machine instruction to be executed next based on the program counter updated by the decoding-executing means.

22. The virtual machine of Claim 21, wherein the decoding-executing means includes a real machine instruction sequence storing unit that stores a plurality of real machine instruction sequences that each correspond to a different virtual machine instruction,

wherein the instruction blocks in the instruction storing means each include a decoded data sequence area for storing a decoded data sequence that specifies real machine instruction sequences in the real machine instruction sequence storing unit, the real machine instruction sequences corresponding to compressed virtual machine instructions stored in the non-branch instruction area and the branch instruction area in the instruction block,

wherein if a decoded data sequence is stored in an instruction block where reading is to be performed, the read means reads a set of decoded data in the decoded data sequence instead of a compressed virtual machine instruction, and if not, the read means reads a compressed virtual machine instruction, decompresses the compressed virtual machine instruction to generate a decompressed virtual machine instruction, and then generates a set of decoded data to specify a real machine instruction sequence corresponding to the decompressed virtual machine

instruction in the real machine instruction sequence storing unit, and

wherein the decoding-executing means reads from the real machine instruction sequence storing unit the real machine instruction sequence specified by a set of decoded data that has been either read or generated by the read means, and executes the real machine instruction sequence.

23. The virtual machine of Claim 22, wherein each instruction block in the instruction storing means further includes a flag area for storing a flag that indicates whether a decoded data sequence is stored in the decoded data sequence area of the instruction block,

wherein the decoding-executing means includes a current flag storing unit for storing a flag that is read from a flag area in a branch destination instruction block by the decoding-executing means when executing a branch instruction, and

wherein the read means reads a set of decoded data or a compressed virtual machine instruction depending on the flag in the current flag storing unit.

24. The virtual machine of Claim 22, wherein each instruction block in the instruction storing means further includes a flag area for storing a flag that indicates whether a decoded data sequence is stored in the decoded

data sequence area of the instruction block and

wherein the decoding-executing means includes a decoded data sequence writing unit for judging, after a branch instruction has been executed, whether the instruction block designated as the branch destination by the branch instruction stores a decoded data sequence by referring to a flag stored in a flag area of the instruction block, and if no decoded data sequence is stored, having a compressed virtual machine instruction sequence in the instruction block read and decompressed, having the decompressed virtual machine instruction sequence decoded to produce a decoded data sequence, and writing the decoded data sequence into a decoded data sequence area in the instruction block.

25. A Just-In-Time (JIT) compiler for use with a virtual machine that executes a virtual machine instruction sequence under control of a real machine, the JIT compiler converting parts of the virtual machine instruction sequence into real machine instruction sequences before execution, and

the JIT compiler comprising:

block start information receiving means for receiving an input of block start information for each virtual machine instruction that composes the virtual machine instruction sequence, the block start information showing whether a corresponding virtual machine instruction would correspond

to a start of a basic block if the virtual machine instruction sequence were divided into basic blocks;

converting means for converting virtual machine instructions in the virtual machine instruction sequence into real machine instruction sequences; and

outputting means for rearranging the real machine instruction sequences produced by the converting means into basic block units in accordance with the block start information received by the block start information receiving means.

26. The JIT compiler of Claim 25, further comprising branch violation judging means for judging, when a real machine instruction at a start of a produced real machine instruction sequence corresponds to a virtual machine instruction whose block start information indicates that the virtual machine instruction would be a start of a basic block, whether the real machine instruction is going to be arranged in an address that violates an address alignment restriction of the real machine,

wherein if the real machine instruction is going to be arranged in an address that violates the address alignment restriction, the outputting means rearranges the real machine instruction sequence so that the real machine instruction is not arranged in the address.

27. The JIT compiler of Claim 26, wherein the outputting means rearranges the real machine instruction sequence by inserting a necessary number of no-operation instructions at the start of the basic block to which the real machine instruction belongs.

28. The JIT compiler of Claim 25, wherein the outputting means inserts a certain number of no-operation instructions at a start of each basic block, the number being a number of real machine instructions processed during a delay of a delayed branch.

29. A storage method used by instruction storing means that stores a virtual machine instruction sequence to be executed by a virtual machine, having a stack architecture, under control of a real machine,

the storage method being characterized by storing each virtual machine instruction in the virtual machine instruction sequence associated with different succeeding instruction information, the succeeding instruction information for a given virtual machine instruction indicating a change in a storage state of data in a stack due to execution of a virtual machine instruction executed after the given virtual machine instruction.

30. A storage method used by instruction storing means

that stores a virtual machine instruction sequence to be executed by a virtual machine under control of a real machine,

wherein the storage method results in:

the instruction storing means being a plurality of instruction blocks that constitute the virtual machine instruction sequence, the instruction blocks corresponding to basic blocks;

the instruction blocks each including:

an identifier area for storing an identifier that specifies a start position of the instruction block in the instruction storing means;

a non-branch instruction area for storing non-branch instructions belonging to a corresponding basic block; and

a branch instruction area for storing at least one branch instruction belonging to the corresponding basic block; and

each branch instruction stored in the branch instruction area designating a branch destination using an identifier stored in one of the identifier areas.

31. A computer-readable recording medium that stores a program to have a computer function as a virtual machine with a stack architecture,

wherein the virtual machine comprises:

stack means for temporarily storing data in a last-in first-out format;

instruction storing means for storing a virtual machine instruction sequence and a plurality of sets of succeeding instruction information, wherein each virtual machine instruction in the virtual machine instruction sequence is associated with a set of succeeding instruction information that indicates a change in a storage state of the data in the stack means due to execution of a virtual machine instruction executed after the associated virtual machine instruction;

read means for reading a virtual machine instruction and an associated set of succeeding instruction information from the instruction storing means; and

decoding-executing means for specifying and executing operations corresponding to a combination of the read virtual machine instruction and the read set of succeeding instruction information.

32. A computer-readable recording medium that stores a program to have a computer function as a compiler that generates a program for a virtual machine with a stack architecture,

wherein the compiler comprises:

instruction sequence converting means for converting

a source program into a virtual machine instruction sequence executable by the virtual machine;

succeeding instruction information generating means for generating sets of succeeding instruction information corresponding to virtual machine instructions in the virtual machine instruction sequence, each set of succeeding instruction information indicating a change in a storage state of data in the stack due to execution of a virtual machine instruction executed after a virtual machine instruction corresponding to the set of succeeding instruction information; and

associating means for associating each set of generated succeeding instruction information with a corresponding virtual machine instruction and outputting the set of succeeding instruction information and the virtual machine instruction.

33. A computer-readable recording medium that stores a program to have a computer function as a virtual machine,

wherein the virtual machine comprises:

instruction storing means for storing a virtual machine instruction sequence;

read means for reading a virtual machine instruction in the virtual machine instruction sequence from the instruction storing means; and

decoding-executing means for specifying and executing

operations corresponding to the virtual machine instruction,

wherein the decoding-executing means includes

a branch instruction judging unit for judging if the virtual machine instruction is a branch instruction and

an interrupt handling unit for detecting, if the virtual machine instruction is judged to be a branch instruction, whether there is an interrupt request, and, if so, performing a corresponding interrupt handling in addition to executing the branch instruction.

34. A computer-readable recording medium that stores a program to have a computer function as a virtual machine,

wherein the virtual machine comprises:

instruction storing means for storing a virtual machine instruction sequence;

read means for reading a virtual machine instruction in the virtual machine instruction sequence from the instruction storing means; and

decoding-executing means for specifying and executing operations corresponding to the read virtual machine instruction,

wherein the decoding-executing means includes

a block judging unit for judging if the read virtual machine instruction is a virtual machine instruction representative of a block, a block being a predetermined number of virtual machine instructions and

an interrupt handling unit for detecting, if the read virtual machine instruction is judged to be the representative virtual machine instruction, whether there is an interrupt request to the virtual machine, and if so, performing a corresponding interrupt handling in addition to executing the representative virtual machine instruction.

35. A computer-readable recording medium that stores a program to have a computer function as a virtual machine,

wherein the virtual machine comprises:

real machine program storing means for storing a plurality of subprograms composed of real machine instructions;

instruction storing means that includes a first area for storing a virtual machine instruction sequence and a second area for storing a plurality of pointers to the subprograms in the real machine program storing means;

read means for reading a virtual machine instruction in the virtual machine instruction sequence from the first area in the instruction storing means; and

decoding-executing means for specifying and executing operations corresponding to the read virtual machine instruction,

wherein the decoding-executing means includes

an area judging unit for judging whether the virtual machine instruction is an instruction that transfers control

flow to a location in the second area and

an address converting-executing unit for executing, if the virtual machine instruction is judged to be an instruction that transfers control flow to a location in the second area, a subprogram indicated by a pointer stored in the location.

36. A computer-readable recording medium that stores a program to have a computer function as a virtual machine,

wherein the virtual machine comprises:

instruction storing means for storing a virtual machine instruction sequence;

read means for reading a virtual machine instruction in the virtual machine instruction sequence from the instruction storing means; and

decoding-executing means for specifying and executing operations corresponding to the read virtual machine instruction,

wherein the instruction storing means is a plurality of instruction blocks that constitute the virtual machine instruction sequence, the instruction blocks corresponding to basic blocks,

wherein the instruction blocks each include: an identifier area for storing an identifier that specifies a start position of the instruction block in the instruction storing means; a non-branch instruction area for storing

non-branch instructions belonging to a corresponding basic block; and a branch instruction area for storing at least one branch instruction belonging to the corresponding basic block,

wherein each branch instruction stored in the branch instruction area designates a branch destination using an identifier stored in one of the identifier areas, and

wherein if the read virtual machine instruction is a branch instruction, the decoding-executing means has control flow branch to a start position of a non-branch instruction area in an instruction block having an identifier designated by the branch instruction as a branch destination.

37. A computer-readable recording medium that stores a program to have a computer function as a virtual machine,

wherein the virtual machine comprises:

instruction storing means for storing a compressed virtual machine instruction sequence to be executed;

read means for reading a compressed virtual machine instruction in the compressed virtual machine instruction sequence from the instruction storing means and decompressing the compressed virtual machine instruction to generate a decompressed virtual machine instruction; and

decoding-executing means for specifying and executing operations corresponding to the decompressed virtual machine instruction,

wherein the instruction storing means is a plurality of instruction blocks containing compressed virtual machine instructions constituting the compressed virtual machine instruction sequence, the instruction blocks corresponding to basic blocks,

wherein the instruction blocks each include: an identifier area for storing an identifier that specifies a start position of the instruction block in the instruction storing means; a non-branch instruction area for storing compressed non-branch instructions belonging to a corresponding basic block; and a branch instruction area for storing at least one compressed branch instruction belonging to the corresponding basic block,

wherein each compressed branch instruction stored in a branch instruction area designates a branch destination using an identifier stored in one of the identifier areas, and

wherein if the decompressed virtual machine instruction is a branch instruction, the decoding-executing means has control flow branch to a start position of a non-branch instruction area in an instruction block having an identifier designated by the branch instruction as a branch destination.

38. A computer-readable recording medium that stores a program to have a computer function as a Just-In-Time (JIT)

compiler used with a virtual machine that executes a virtual machine instruction sequence under control of a real machine, the JIT compiler converting parts of the virtual machine instruction sequence into real machine instruction sequences before execution,

wherein the compiler comprises:

block start information receiving means for receiving an input of block start information for each virtual machine instruction that composes the virtual machine instruction sequence, the block start information showing whether a corresponding virtual machine instruction would correspond to a start of a basic block if the virtual machine instruction sequence were divided into basic blocks;

converting means for converting virtual machine instructions in the virtual machine instruction sequence into real machine instruction sequences; and

outputting means for rearranging the real machine instruction sequences produced by the converting means into basic block units in accordance with the block start information received by the block start information receiving means.